

# LEADING by EXAMPLE: THE CASE FOR IT SECURITY in ACADEMIA

By Mary Ann Davidson

About five or six years ago, many of us in IT security felt like the Maytag repairman: the phone never rang; no one called, no one wrote; apparently, no one cared. The only people who were concerned about security were the “professional paranoids”—defense, intelligence, and the occasional financial institution.

What changed? Well, “Internet happens.” Actually, Internet had already happened, but people started using it. They wanted to do business with customers and partners, they wanted to share information more widely, and all of a sudden, we had a disappearing or porous network perimeter. Security has become something everyone has to worry about.

Security has also become a business issue: all elements of the critical infrastructure—government, telecommunications, financial institutions—now have an IT backbone. It has to be as safe, secure, and reliable as physical infrastructure, and it isn't, yet. Furthermore, the

network is the battlefield, so to speak. Attackers need find only one hole or one way in, but defenders have to prevent all holes and protect all ways in. It's an unequal battle, and the collusion of the bad guys is better than the collective defense of the good guys.

Leadership in IT security is needed. Security matters: the ethics, the economics, and the social implications. There is much the academic community can do to help ensure cybersecurity.

## The Ethics of Security

There's a story in my family lore about a couple of dear elderly ladies, Emily and Lucille. Emily was in her seventies, and Lucille was in her eighties. One day Emily told Lucille that she was thinking about getting a facelift. Lucille looked at her and said, soberly, “Emily, it's too late.”

What's the point to this story? When seventeen- and eighteen-year-old students arrive at academic institutions, their ethical

beliefs have largely been formed. If these students do not already understand the importance of ethics or responsible behavior, it will be very difficult for an academic institution to instill it in four years. At least once a year, there is an article in a business publication about why MBAs don't seem to practice ethical behavior. The answer is that it's too late at that point, and the schools don't select for ethics, either. If business schools select people who value money over everything else, they're not going to be able to teach these students, in two short years, that doing the right thing is more important than making money.

That does not mean that colleges and universities should give up and not do anything to foster ethics and integrity. Ethical behavior—as it pertains to computing, to the use of computing resources—needs to be a part of the curriculum. Many colleges and universities have codes of conduct covering issues such as plagiarism, hacking, snooping, piracy, and file-sharing. The obvious question is: Are they enforced? There are lots of excuses for bad behavior today, and there is very little accountability.

*Mary Ann Davidson is Oracle's Chief Security Officer, a post that she argued to formally create and was appointed to in 2001. As Oracle CSO, she is responsible for product security, security evaluations, and security assessments.*



I'd like to tell a couple of stories about why ethics matter. The first is the story behind the SQL Slammer worm. There is a very technically savvy security "researcher" who likes to present papers at Black Hat, a hacker conference. Several years ago he delivered a paper about a vulnerability he had found in Microsoft software and had reported to Microsoft. (The company fixed the problem very quickly and provided a patch.) In his Black Hat paper, the researcher explained what bad things could happen to customers who did not apply the patch, and he handed out the exact exploit code enabling *anybody* to do the bad thing. A couple of months later, the Slammer worm came out, and the basis for the code was the researcher's Black Hat paper. The researcher didn't write the worm, but he made it possible for someone else to do so. The researcher stated, in justification, that he was just trying to protect Microsoft customers. However, my response is: "Wait a minute. If you really wanted to do that, you should have presented that paper at the Microsoft Users Group conference. Not at Black Hat. You know who the audience is at Black Hat. They're hackers." The researcher also said, "Well, anyone else could have written what I wrote in twenty minutes." My response: "No. 'Anyone else' didn't write it. You took the knowledge that you had and made it possible for anyone else to do something bad. That's wrong." I don't think this behavior is ethical; it's not illegal, but it is wrong.

Another ethics story involves insider information on security bugs. I am occasionally asked—for example, by the U.S. Department of Homeland Security, as part of a partnership that it is putting together—about giving advance notice on security bugs, meaning that some customers would know about vulnerabilities before others. My response is, "Absolutely not." Why? Because every one of my customers would want the same thing, and about 95 percent of my customers are in critical infrastructure. I treat everyone, even my own IT department, exactly the same. I don't think one group of customers should have an advantage over another. Plus, 30,000 people can't keep a secret. The moral principle at stake says that you should not put one group of customers at

greater risk than others (or leave one group of customers at a disadvantage relative to others).

There are exceptions. Cisco had a vulnerability that it felt would affect the health of the Internet, so the company contacted backbone providers first to ensure their systems were patched. That's a good exception case. Or if you find a protocol fault that affects everyone, of course you should go to the U.S. Computer Emergency Response Team (CERT) and say, "Let's all work together—all the vendors should be patching this at once." But in general cases, advance notice or insider information on security vulnerabilities is unethical.

There are people who create businesses out of vulnerability trading. For example, there are firms that buy vulnerabilities from hackers and sell them on a subscription basis. Mind you, they don't vet their customer base, meaning that—for all we know—terrorists or organized crime rings could be happily buying this information. Furthermore, by "outing" vendors over specific vulnerabilities, these firms force a vendor to fix the issue they bought from hackers, whether or not there are other, more critical security issues the vendor is working on (issues that the "insider trading" firm does not know about).

I admit: We vendors need to do better in building secure software. We try. I try everything I possibly can. But I don't buy the argument that it's all the vendors' fault for not building better software, in cases where researchers happily hand weapons of mass disruption to the hacking community. That's like saying: "I have a Molotov cocktail. I'm going to leave it on the front lawn of a building with a 'throw me' sign on it, and when the building burns down, I'm going to say, 'I just wanted better building codes.'" Tell that to the people who just lost their house. It's wrong. With knowledge comes responsibility. Clever researchers or hackers who find ways to break things should use that knowledge to build more defensible systems, not to burn down cyberhouses.

Clever researchers or hackers who find ways to break things should use that knowledge to build more defensible systems, not to burn down cyberhouses.



## The Economics of Security

Economic principles need to be applied to IT security. There is a concept in economics called *social cost*. For example, a byproduct of manufacturing may be air pollution, which results in higher lung cancer rates. The manufacturer does not "pay" by installing smokestack scrubbers, yet we all "pay" through bad health, higher emphysema rates, and so on.

One could argue that the costs associated with product security vulnerabilities also constitute a social cost. Customers pay by having to apply a lot of patches (expensive in terms of both manpower and system downtime), and they

may also pay by being attacked by a worm or virus that exploits an unpatched vulnerability. Yet, customers lack the information to be able to make better purchasing decisions in which they know the all-in "cost to secure" before buying a product.

There are multiple economic analyses that are useful to apply to software security, and more research could be done here. Economics can help "make the case" for IT security.

*Cost avoidance.* As justification for the popular strategy of "throw the product over the wall as fast as possible to gain market share at all costs," many vendors say that nobody will pay more for more secure software. For example, we know exactly what it costs us as a company to produce a patch for a software fault that's been out there for a long time and that may occur on all product versions, on all operating systems. In the worst case, we have done seventy-eight patches for one vulnerability. When you look at the amount of money spent fixing avoidable, preventable security faults, you realize that almost anything that can be done upfront to avoid making those mistakes pays for itself quickly. There is a strong cost-avoidance argument to building software right the first time. Also, there is an opportunity cost argument. Imagine the features you could build (and thus the greater number of products you could sell) with the same resources you use to fix avoidable, preventable security bugs after the fact.

*Expected value.* When you look at the economics of how good is “good enough security,” you have to factor in expected value. In other words, if the customers can’t keep up with patches—if they miss one and get whacked by a worm or a virus—that’s a negative payoff.

*Return on investment.* Better security often results in lower cost. For example, Oracle had an intrusion-detection system that we recently replaced with a system provided by another vendor. We analyzed the value and accuracy of the information provided by each system. The old system had a ridiculously high number of alarms every week, and an extraordinary amount of them—70 to 80 percent—were false positives. We looked at what it was costing us to track down the alarms that we really needed to do something about, including the costs for people to sort through the alarms and analyze them. The new product had a much lower alarm rate as well as a lower false positive rate. The information provided by the new product was better, at a lower cost. Economic analysis, specifically return on investment, helped us choose the new supplier over the old one.

One of the few areas in security in which you can show a clear return on investment is single sign-on. Many companies have large help-desk expenses due to password resets (“I forgot my password; can you please reset it?”). This trend has only accelerated as companies roll out more self-service applications. Deploying a single sign-on system can pay for itself in months, not years. This is one area where you can justify spending money on security because it results in measurable benefits. Furthermore, single sign-on, as part of a larger identity-management system, may help you deploy applications faster, improving time-to-market.

### The Social Implications of Security

Often, people will get so wrapped up in technology that they’ll do something simply because they can. As a society, we

Technology is simply too important to be left to so-called technical experts, many of whom become enamored with technology for technology’s sake.



should discuss the ramifications of technology before it becomes ubiquitous.

Data aggregation has interesting implications, good and bad. A company in Las Vegas has a profitable business determining who is who and who is related to whom. Why would someone care about this in Las Vegas? Because by law, casinos are not allowed to do business with certain people, and the Nevada State Gaming Commission maintains a database of who those people are. When you check into a hotel in Las Vegas, the first thing the hotel may do is check your name and variants of your name against this database. If I go in and say that I’m Mary Ann Davis, they’ll prob-

ably figure out that “Davis” is similar to “Davidson,” and they’ll know who I am if I’m in that database. Casino employees can also use software to figure out who is related to whom. For example, they can determine if the person who just won a big hand at the poker table has a relative at the same address as the dealer. Data aggregation is thus useful to help prevent crime such as fraud or insider collusion.

Data aggregation has interesting implications and applications in the intelligence community. Consider the practice of “customer profiling.” Each of us has a digital fingerprint, and our transactions tell a story about who we are, our likes and dislikes. Let’s say you’re at work and you get an e-mail from Amazon saying, “Based on your last five book purchases, we think you might really like this new book.” You may think: “Hey, that’s a great service. They really know me. I’m going to check out that book.” But if, on the other hand, you receive an e-mail from the Federal Bureau of Investigation saying, “Based on your last five book purchases, we think you might be a security risk,” you’d likely be more concerned. The point is: both organizations use the same information. In the first case, you think it’s a nice service that Amazon knows so much about you, but in the second case, you’re incensed that the FBI knows so much about you. Yet, it is the same data. In some cases, so much infor-

mation is publicly available that you can find out a lot about people just by doing a good Web crawl (where they live, what they paid for their house, where their kids go to school, and so on).

I modestly propose that there is a Law of Conservation of Data: Once data is collected, it is never destroyed; it lives forever. Its corollary is the Law of Unintended Data Usage: Once data is collected for one purpose, the temptation to use it for another purpose is too great to resist. For example, the U.S. Armed Forces several years ago started taking DNA samples for the ostensibly noble purpose of wanting never again to have a Tomb of the Unknown Soldier. Hence, the military began taking DNA samples: a service member’s remains could thus always be identified against his or her stored DNA. Many service members balked at the requirement, however; they wanted assurance that the DNA would never be used for other purposes (such as unsolved crimes, paternity cases, and so on).

A final example of IT use with social implications is electronic voting. As a security person, even though I don’t have a problem with electronic voting per se, I have serious problems with someone saying, “The security mechanisms of our voting machines are secret, but you should all trust us that we did our jobs properly.” Oracle pays a lot of money to have outside firms validate the security of our products. Why? Because when it comes to security, the mantra should be “trust, but verify.” Having someone *other than* the vendor attest to the security of the product is standard practice in the IT industry (ISO-15408, the International Common Criteria). The “trust, but verify” issue is particularly significant when we’re talking about voting systems in a democracy. All citizens need to know that the system has been vetted if we are to trust our futures to it.

### What Can Academia Do?

What steps can academia take to help ensure cybersecurity? First, higher education institutions need to foster nerdy liberal arts majors. Technology is simply too important to be left to so-called technical experts, many of whom become enamored with technology for technology’s sake. Liberal arts majors are among those who are likely to go into politics and craft

future legislation. It is thus imperative that these people—the liberal arts majors—understand technology: they need to know what they’re legislating. If they don’t, they’ll be looking to someone else to define their technical roadmap.

Likewise, academia should help IT students to become well-rounded nerds. One of the many things I loved about the University of Virginia, where I received my bachelor’s degree in engineering, was that the university had (and still has) a Humanities Division in the Engineering School. I couldn’t graduate in engineering without reading Plato’s *Republic* and Sir Thomas More’s *Utopia*. We talked about the social implications of technology. The people who are building technology need to gain this broad perspective.

Academia should also help shape the field. Computer science is not yet a profession in the sense that engineering is. Consider this analogy: what if civil engineers built bridges the same way that developers write software? Are our networks and systems as reliable as our bridges and buildings? No. Civil engineers know what’s at

stake: they focus on making things safe, secure, and reliable, and an engineer knows that if a bridge falls down, he or she is responsible.

IT is now infrastructure, as well. It needs to be as safe, secure, and reliable as physical infrastructure, but it isn’t. In the software industry, nobody is responsible for failure and nobody is liable. If a program fails due to poor design or execrable coding practice, no one stands up and says, “I’m accountable.” If a bridge fails, maybe not everyone who worked on that bridge is accountable, but *somebody* is, such as the licensed professional engineer (PE) who signed off on the structural drawings.

Software developers say: “If you require more rigorous development processes with all these extra requirements for security, it’ll stifle my creativity.” My response is: “Architects are very creative, but they’re creative within the bounds of sound structural engineering, building codes, cost, location, and client preferences.” Or developers may say: “Software is complex. It’s too complicated to expect that level of safety and reliabil-

ity.” Well, so is the power grid. Yet the power grid seems to be reliable. Don’t tell me that we can’t make software that is as reliable as the power grid.

IT security is fundamentally a cultural issue. Engineers would never think about “coolness” or “elegance” as being more important than safety, security, and reliability. This is a cultural transformation that computer science needs to make.

Oracle has a rather strong culture of security, partly because we started as a company nearly thirty years ago by building the first commercial relational database, under contract to a U.S. intelligence agency. However, security needs to be a red-button issue for everyone involved in information technology: if something isn’t secure enough, you push the button and stop the assembly line. Academia should foster this culture of security—not only by developing the computer science curriculum and shaping the computer science profession, but also by protecting poorly defended machines, implementing acceptable use policies, aligning with security standards, giving authority to college

and university IT security professionals, conducting routine penetration tests, providing awareness training, and reviewing logs regularly.

*Protect poorly defended machines.* Several things can be done to defend your turf as well as give you more control at lower cost—always attractive to management in any organization.

To the extent practical, and this may be harder to do in an academic institution than in a corporation, having standard machine configurations makes it easier to determine what a “secure” configuration is and to enforce it. Obviously, using automated tools to check the security settings helps as well. Appropriately segregating the networks is another defense mechanism. Even allowing for broad idea exchange and academic freedom, does every student need to be able to access all parts of the network?

Consolidating systems where it makes sense to do so is another way to improve security. For example, Oracle used to have several dozen mail servers around the world, with a server in virtually every country in which we did business. Need-

less to say, not all the mail administrators in each of those countries were equally proficient at tuning the system and maintaining it. To save money and improve the quality of mail service, we started to consolidate our systems (using our own mail server product at the back end). We did several rounds of regional consolidation first, before consolidating to three mail instances worldwide.

In addition to better control and much lower cost, we also realized a security benefit. For example, when the Melissa virus came through, the antivirus software we had at the mail gateway didn't immediately have the virus signature for it. However, our mail team (which was off-site) dialed into the mail servers and deleted all the “Melissa” messages, so none were delivered. We were done with Melissa in about twenty minutes, by effectively “inoculating” our consolidated mail servers. (Another company in Silicon Valley took about twenty-seven people and three weeks to clean up after Melissa.) Having a few machines that are well-defended is better than having 800 machines that are

impossible to defend, and fewer machines are cheaper to manage too.

*Implement acceptable-use policies.* An organization that has an acceptable-use policy can go to users (if they are misbehaving) and say: “Did you read this? We know you did because you had to read it and sign it. You didn't use the machine resources that we gave you for acceptable use.” The idea of acceptable use can mean anything from not hacking to not having porn on a computer. Whatever the acceptable use, if you make it policy and have people sign off on it, then you've got something to hold them accountable to.

*Align with security standards.* We are currently aligning parts of our organization with ISO 17799. ISO 17799 is generally good security practice, and the more an organization aligns with this standard, the more it can prove that it is doing the good things it should be doing anyway. Many of our customers are asking that of us, especially those who are outsourcing management of their business applications to us.

*Give authority to IT security.* Do your security specialists have enough authority to

say, “No, we’re not doing this”? About a year ago, we restructured security. Even before then, I had plenty of authority. My focus was mostly product security, and I was one of a handful of people in the company who could stop a product release. But my counterparts who were responsible for global information policies and physical security were buried too deep in the organization. People in other lines of business would occasionally say: “Oh, you’re the security people. We don’t have to do what you say.” That’s not true anymore. We all now report to the Chief Corporate Architect, who reports to our CEO. So all of a sudden, we are all two levels below the chief executive, and people return our phone calls. The people making security policy have the authority to enforce it. Responsibility without authority equals frustration.

*Conduct routine penetration tests.* My ethical hacking team is very good at what it does. The reason it does what it does—attempt to break our products and attempt to break into our networks—is to make sure that we’re keeping up with the latest hacking techniques so that our product security

and practice of security remain robust. Development teams also want their code to be hacked. They like the challenge, and they know the process is saving them pain and agony in the long run. “Try to break your own security before hackers do” is good practice.

*Provide awareness training.* Ignorant end users exacerbate many security problems. When the IT department sends a message that says, “You need to update, you need to run your virus definitions, you need to apply this patch,” users should respond. Once IT says that something is urgent, I stop whatever I’m doing—I don’t care what it is—and do whatever they tell me to do. Institutions need to instill that behavior in their users. Awareness training will help make everyone’s job easier. For example, users need to be trained to recognize social engineering, such as phishing attacks: “We’re from your bank,

Customers want something that’s safe, secure, and reliable, and this needs to be the *main* concern of computer science today.



and we want you to confirm your account number again via e-mail.” It’s not merely true that “on the Internet, nobody knows you’re a dog.” It’s also true that “on the Internet, nobody knows you are a low-down *dirty* dog.”

*Review logs regularly.* When it comes to reviewing logs, most people say, “I’ve got audit logs, and I turn the firewall logging on, and I’ve got entries in the intrusion detection system.” But they never look at the logs, partly because they’re so overwhelmed with this big pile of data. Fortunately, there are products that will take all the logs, do data warehousing from the data, and draw infer-

ences for you, such as, “You might want to go investigate this or that.” These products are another way that you can be smarter with the data you already have, and that helps you do more with less.

## Conclusion

Academia has a critical role to play in leading the effort to secure cyberspace. I don’t like having to organize secure coding classes. I don’t like having to explain to developers: “Hey, you need to think about the business ramifications of this. You need to think about the cost of bad security. You need to think about really good programming practice.” I think they should have learned all that in academic institutions. Of course, it’s entirely possible that developers did learn all this but that what they learned was stomped out of them by the IT industry.

Academia can also help change the focus of computer science majors and of the computer science profession. Customers want something that’s safe, secure, and reliable, and this needs to be the *main* concern of computer science today. Finally, academia can help non-techies become as technically literate as possible. Technology is here, and it has implications that are, to me, a little frightening. As a society, we should be discussing the ethics, economics, and social implications of IT and of IT security. There is no better place for that discussion to start than academia. *e*